

# MCMC inference for Markov Jump Processes via the Linear Noise Approximation

Vassilios Stathopoulos\* and Mark A. Girolami†  
 Department of Statistical Science,  
 Centre for Computational Statistics and Machine Learning,  
 University College London, Gower Street, London WC1E 6BT, UK

November 2012

## Abstract

Bayesian analysis for Markov jump processes is a non-trivial and challenging problem. Although exact inference is theoretically possible, it is computationally demanding thus its applicability is limited to a small class of problems. In this paper we describe the application of Riemann manifold MCMC methods using an approximation to the likelihood of the Markov jump process which is valid when the system modelled is near its thermodynamic limit. The proposed approach is both statistically and computationally efficient while the convergence rate and mixing of the chains allows for fast MCMC inference. The methodology is evaluated using numerical simulations on two problems from chemical kinetics and one from systems biology.

## 1 Introduction

Markov Jump Processes (MJP) provides us with a formal description of the underlying stochastic behaviour of many physical systems and as such they have a wide applicability in many scientific fields. In chemistry and biology, for example, they are applied for modelling reactions between chemical species [1, 2]. In ecology and epidemiology, they are used for modelling the population of interacting species in the environment [3] while in telecommunications they describe the population of information packets over a network [4]. In order to introduce some terminology and notation we will give a more concrete example from chemical kinetics. However, the modelling methodology is similar in other applications although different assumptions are needed, depending on the system being modelled, for calculating reaction rates. Consider a model for the population of molecules of two interacting chemical species,  $X_A$  and  $X_B$ , in a solution of volume  $\Omega$ , where  $X_A$  and  $X_B$  denote the number of molecules of chemicals  $A$  and  $B$  respectively. The interactions between the species are modelled using *reactions* which are specified using the following notation:  $R_1 : A + B \xrightarrow{c_1} 2A$ . On the left hand side appear the *reactants* and on the right hand side the *products* of the reaction while over the arrow appears the *rate constant*  $c_1$  which is the probability that a randomly chosen pair of  $A$  and  $B$  will react according to  $R_1$ . This reaction, for example, specifies that a pair of molecules  $A$ ,  $B$  react with probability  $c_1$  to produce a new molecule of  $A$ . For calculating the probability of a reaction taking place given the current state of the system, *i.e.* the number of molecules of chemicals  $A$  and  $B$ , several system dependent assumptions must be made. For chemical reactions it is assumed that in a well stirred solution the probability of a reaction is proportional to the populations of its products [5]. For  $R_1$  we can write it as  $f_1(X_A, X_B, c_1) = c_1 \Omega^{-1} X_A X_B$ . Following the same reasoning additional reactions and species can be added in order to construct large and complex reaction networks.

---

\*1 v.stathopoulos@ucl.ac.uk

†2 m.girolami@ucl.ac.uk

Together, the state of the system  $X_A, X_B$ , the set of reactions and the reaction rates specify a Markov Jump process where the occurrences of reactions are modelled as a Poisson process.

For this particular example the probability of the reaction has a simple form and is linear with respect to the populations. However in many real applications this is often not the case while the *rate constants*,  $c_1$ , are unknown. Given a fully specified MJP, *i.e.* a MJP with known parameters, rate constants and initial conditions, it is possible to perform exact simulation and obtain samples from the underlying stochastic process using the Stochastic Simulation Algorithm (SSA) of [1]. In many problems there are system parameters which are not specified or are unknown while it is relatively easy to collect partial observations of the physical process at discrete time points. The interest is therefore to obtain statistical estimates of the unknown parameters using the available data.

As a consequence of the Markov property, MJPs satisfy the Chapman-Kolmogorov equation from which we can directly obtain the forward master equation describing the evolution of the system's state probability over any time interval. However, even for small and simple systems the master equation is intractable and it is not straightforward as to how partially and discretely observed data from the physical process should be incorporated in order to perform inference over unknown system parameters. Recently, [6] have shown that it is possible to construct a Markov Chain whose stationary probability distribution is the posterior of the unknown parameters without resorting to any approximations of the original MJP. Their method however is computationally expensive while the strong correlation between posterior samples means that a large number of MCMC iterations are required in order to obtain Monte Carlo estimates with sufficient accuracy.

An alternative is to consider suitable approximations of the likelihood function. The system size expansion of [7, Chap. 10] provides a systematic method for obtaining approximations of a physical process approaching its thermodynamic limit. The most simple approximation yields the Macroscopic Rate Equation (MRE) which describes the thermodynamic limit of the system with a set of ordinary differential equations neglecting any random fluctuations. Although the MRE has been extensively studied in the literature, see for example, [8, 9], it is not applicable for problems where information about the noise and the random fluctuations is necessary or the system is far from its thermodynamic limit. The diffusion approximation [10, 11] describes the physical process by a set of non-linear stochastic differential equations with state dependent Brownian motion. Similar to the master equation however, the likelihood is intractable. In [12] a transformation is applied such that the Brownian increments are independent of the system state and thus the system can be easily simulated. However this limits the applicability of the methodology into systems where such a transformation is possible. A more general methodology is presented in [13] where an approximation of the likelihood is used instead. Finally, a less studied approach for the purpose of inference is the Linear Noise Approximation (LNA) which conveniently decouples non-linearity in the diffusion approximation into a non-linear set of ordinary differential equations in the MRE and a set of linear stochastic differential equations for the random fluctuations around a deterministic state [7, Chap. 10],[14]. Recently, [15] have shown the simple analytic form of the approximate likelihood obtained by the LNA simplifies MCMC inference and can be applied to problems with relatively small number of molecules.

A commonly employed algorithm for MCMC is the Metropolis-Hastings algorithm [16], which relies on random perturbations around the current state using a local proposal mechanism. It should be noted here that the state of the Markov chain is different from the state of the stochastic process. In the MCMC context state refers the current values of the unknown system parameters whereas the state of the system refers to the value of the stochastic process at a given time. We will use the term state interchangeably for the rest of this paper and its meaning will be clear from the context. Due to the local nature of the proposal mechanism used by the Metropolis-Hastings algorithm, samples from the posterior exhibit strong random walk behaviour and auto-correlation. Tuning the proposal mechanisms to achieve good mixing and fast convergence is far from straightforward even though some theoretical guidance is provided

[17]. MCMC methods, such as the Metropolis Adjusted Langevin Algorithm (MALA) [18] and the Hamiltonian Monte Carlo (HMC) [19], have also been studied in the literature and have been shown to be more efficient than random walk Metropolis-Hastings in terms of Effective Sample Size (ESS) and convergence rates on several problems. However, HMC and MALA also require extensive tuning of the proposal mechanisms, see for example [20] and [21]. For MJPs the problem is compounded further since system parameters, such as probability rate constants of chemical reactions, are often highly correlated and whose values may differ by orders of magnitudes. The resulting posterior distributions have long narrow “valleys” preventing any local proposal mechanism from proposing large moves about the parameter space.

More recently [22] proposed exploitation of the underlying Riemann manifold of probability density functions when defining MCMC methods thus exploiting the intrinsic geometry of statistical models, thereby providing a principled framework and systematic approach to the proposal design process. These algorithms rely on the gradient and Fisher Information matrix of the likelihood function to automatically tune the proposal mechanism such that large moves on the parameter space are possible and therefore improve convergence and mixing of the chains. In [9] this approach has been successfully applied for the MRE approximation of chemical reaction networks. For the LNA the Fisher Information and the gradient of the likelihood function can be easily obtained [2]. In this paper we study the application of the Riemann manifold MCMC methods for the LNA approximation and compare the mixing efficiency and computational cost with to the commonly used Metropolis-Hastings algorithm. Moreover we study how the the Markov chains and the resulting Monte Carlo estimates behave for systems which are far from their thermodynamic limit. The aim is to improve the efficiency of MCMC inference for MJPs in order to allow for larger and more complex models frequently encountered in biology and chemistry to be studied in more detail.

In the next section we give a brief overview of Markov jump processes. The diffusion and linear noise approximations are presented in section 3. We then discuss MCMC and the Riemann manifold algorithms in section 4. Numerical simulations are presented in section 6 while section 7 concludes the paper.

## 2 Markov Jump Processes

A  $D$ -dimensional stochastic process is a family of  $D$  random variables  $\mathbf{X}(t) = [X_1(t), \dots, X_D(t)]^T$  indexed by a continuous time variable  $t$  with initial conditions  $\mathbf{X}(t_0) = \mathbf{x}_{t_0}$ . A Markov Jump Process (MJP) is a stochastic process satisfying the Markov property such that

$$p[\mathbf{X}(t_0), \dots, \mathbf{X}(t_N)] = p[\mathbf{X}(t_0)] \prod_{i=1}^N p[\mathbf{X}(t_i) | \mathbf{X}(t_{i-1})],$$

where the dependence on any parameters or other quantities has been suppressed. That is, the conditional probability of the system state at time  $t_i$  only depends on state of the system at the previous time  $t_{i-1}$ . A MJP is characterised by a finite number,  $M$ , of state transitions with rates  $f_j(\mathbf{x}, \boldsymbol{\theta}, t)$  and state change vectors  $\mathbf{s}_j = (s_{1,j}, \dots, s_{D,j})^T$  with  $j \in [1, \dots, M]$ .  $f_j(\mathbf{x}, \boldsymbol{\theta}, t)dt$  is the probability, given the state of the system at time  $t$ ,  $\mathbf{X}(t) = \mathbf{x}$ , of a jump to a new state  $\mathbf{x} + \mathbf{s}_j$  in the infinitesimal time interval  $[t, t + dt)$ . For the problems we consider in this paper the transition rates not only depend on the current state and time but also on unknown rate parameters  $\boldsymbol{\theta}$ . From the Markov property we can directly obtain the conditional probability of the system being in state  $\mathbf{x}$  at time  $t$  given initial conditions which is characterised by the master equation

$$\frac{dp(\mathbf{x}, t | \mathbf{x}_0, t_0)}{dt} = \sum_{j=1}^M [f_j(\mathbf{x} - \mathbf{s}_j, \boldsymbol{\theta}, t)p(\mathbf{x} - \mathbf{s}_j, t | \mathbf{x}_0, t_0) - f_j(\mathbf{x}, \boldsymbol{\theta}, t)p(\mathbf{x}, t | \mathbf{x}_0, t_0)]. \quad (1)$$

Equation (1) in general form is intractable especially when the transition rate functions  $f_j(\cdot)$  are nonlinear with respect to the system state. Numerical simulation is also prohibitively expensive as the computational cost grows exponentially with  $D$  [23].

However, given initial conditions  $\mathbf{X}(t_0) = \mathbf{x}_{t_0}$  and values for the unknown rate parameters  $\boldsymbol{\theta}$  we can simulate realisations of the MJP by first noting that the time  $\tau$  to the next state transition is exponentially distributed with rate  $\lambda = \sum_{j=1}^M f_j(\mathbf{x}_{t_0}, \boldsymbol{\theta}, t_0)$  and the new state  $\mathbf{X}(t_0 + \tau)$  will be  $\mathbf{x}_{t_0} + \mathbf{s}_j$  with probability  $f_j(\mathbf{x}_{t_0}, \boldsymbol{\theta}, t)/\lambda$ . This results in an iterative algorithm from which we can forward simulate a complete trajectory for the stochastic process  $\mathbf{X}(t)$ , known as the Stochastic Simulation Algorithm (SSA) [1] in the chemical kinetics literature.

From the specification of the MJP we can also write the likelihood function with respect to the parameters  $\boldsymbol{\theta}$  for a completely observed process  $\mathbf{X}(t)$  at the time interval  $[0, T]$  as

$$p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{i=1}^N f_{k_i}(\mathbf{x}_{i-1}, \boldsymbol{\theta}, \tau_{i-1}) \exp \left( -\tau_i \sum_{j'=1}^M f_{j'}(\mathbf{x}_{i-1}, \boldsymbol{\theta}, \tau_{i-1}) \right)$$

where  $N$  is the number of transitions occurred in the time interval  $[0, T]$ ,  $k_i \in [1, \dots, M]$  is the type of the  $i^{th}$  transition and  $\tau_i, \mathbf{x}_i$  are the time and state at the  $i^{th}$  transition respectively. Notice that the likelihood function corresponds to the generative process described by the SSA. By specifying a suitable prior and applying Bayes' theorem, we can obtain the posterior distribution  $p(\boldsymbol{\theta}|\mathbf{X})$  which we can use for inference over the unknown parameters  $\boldsymbol{\theta}$  [6].

In many problems of interest however we cannot observe the times and types of all transitions in a given time interval. Rather, we can only observe the state of the system  $\mathbf{X}(t_i) = \mathbf{x}_i$  at discrete time points  $t_i \in [0, T]$ . The solution proposed in [6] is to treat the trajectories, as well as the number, times and types of transitions, between observed time points as latent variables. This leads to a data augmentation framework [24] where a Markov Chain is constructed to sample from the joint posterior of the parameters and the latent variables. At each MCMC iteration the complete trajectory of the MJP process has to be simulated conditional on the observed data and the parameters which for some systems can be computationally demanding. Furthermore, due to the high dimensional nature of the simulated trajectory and the strong dependence on the system parameters and observed data the MCMC algorithm has very poor convergence and mixing properties requiring many samples from the posterior in order to obtain sufficiently accurate Monte Carlo estimates. Finally, a further complication that arises is that the number of transitions between two observed time points is also unknown and has to be sampled using a reversible-jumps type algorithm [25]. For more details see [6]. The resulting algorithm therefore is computationally demanding thus limiting its applicability on small and relatively simple MJPs. A more efficient version of the algorithm is also suggested in [6] where instead of simulating the trajectories between observations using the exact MJP an approximate proposal distribution is employed to sample trajectories which are accepted or rejected using the Metropolis-Hastings ratio.

### 3 Diffusion and Linear Noise Approximations

An alternative to working directly with the master equation and the original MJP is to consider approximations which provide for efficient simulation and possibly an easy to evaluate likelihood function for discretely observed data. Although the resulting posterior will also be approximate in nature, it can be sufficient for inferential purposes given that the system under consideration is near its thermodynamic limit. Here we describe the diffusion approximation and from that how we can arrive at the LNA. Our presentation is rather informal and follows [14] and [1]. For a more formal derivation the reader should refer to [7] and [11]. The requirement for these approximations to be consistent is the existence of a proportionality constant  $\Omega$  which governs the size of the fluctuations such that for large  $\Omega$  the jumps will be relatively small and as both  $\Omega$  and  $\mathbf{x}$  tend to infinity approaching the system's thermodynamic limit then,

$$f_j(\mathbf{x}, \boldsymbol{\theta}, t) \rightarrow \Omega \tilde{f}_j(\mathbf{z}, \boldsymbol{\theta}, t), \quad (2)$$

where  $\mathbf{z} = \mathbf{x}/\Omega$  and  $\tilde{f}_j(\cdot)$  are independent of  $\Omega$ . For many physical processes where the fluctuations are due to the discrete nature of matter there is a natural  $\Omega$  parameter with such

properties. Examples of such parameters can be the system size in chemical kinetics, the capacity of a condenser in electric circuits or the mass of a particle [7].

### 3.1 Diffusion approximation

In order to obtain a Langevin equation which closely matches the dynamics of the MJP it is assumed that there is an infinitesimal time interval  $dt$  which satisfies the following conditions

$$f_j(\mathbf{x}_{t'}, \boldsymbol{\theta}, t') \approx f_j(\mathbf{x}_t, \boldsymbol{\theta}, t), \quad \forall t' \in [t, t + dt), \forall j \in [1, M] \quad (3)$$

$$f_j(\mathbf{x}_t, \boldsymbol{\theta}, t)dt \gg 1 \quad \forall j \in [1, M]. \quad (4)$$

The first condition constrains  $dt$  to be small enough such that the transition rate functions remain approximately constant. This implies that the number of transitions of type  $j$  is distributed as a Poisson random variable with mean  $f_j(\mathbf{x}_t, \boldsymbol{\theta}, t)dt$  and is independent from other transitions of type  $j' \neq j$ . The second condition constrains  $dt$  to be large enough such that the number of transitions for each state is significantly larger than 1, which further implies that the Poisson distribution can be accurately approximated by a Gaussian distribution. It can be shown [26] that we can choose  $dt$  and  $\Omega$  such that both conditions can be satisfied and this generally occurs when the system approaches its thermodynamic limit.

Given such a timescale, the state of the system at time  $t + dt$  can be computed by

$$\mathbf{x}_{t+dt} = \mathbf{x}_t + \sum_{j=1}^M \mathcal{N}[f_j(\mathbf{x}_t, \boldsymbol{\theta}, t)dt, f_j(\mathbf{x}_t, \boldsymbol{\theta}, t)dt] \mathbf{s}_j \quad (5)$$

where  $\mathcal{N}[\mu, \sigma^2]$  denotes a Gaussian random variate with mean  $\mu$  and variance  $\sigma^2$ . From Equation (5) we can directly obtain a Langevin equation of the form

$$d\mathbf{x}_t = \mathbf{S} \mathbf{f}(\mathbf{x}_t, \boldsymbol{\theta}, t)dt + \mathbf{S} \sqrt{\text{diag}[\mathbf{f}(\mathbf{x}_t, \boldsymbol{\theta}, t)]} d\mathbf{B}_t \quad (6)$$

where we used  $\mathbf{S}$  to denote the matrix whose columns are the state change vectors  $\mathbf{s}_j$ ,  $\mathbf{f}(\cdot)$  to denote the vector whose elements are the transition rates  $f_j(\cdot)$ ,  $\text{diag}(\mathbf{v})$  a function that returns a diagonal matrix with elements taken from the vector  $\mathbf{v}$  and  $d\mathbf{B}_t$  an  $M$  dimensional Wiener process. Notice that the dimension of  $\mathbf{x}_t$  differs from that of  $d\mathbf{B}_t$ .

Due to the nonlinear state dependent drift and diffusion coefficients in Equation (6) the transition density of the stochastic process is also intractable. Therefore a data augmentation approach similar to the one in [6] has to be followed. However, there is no longer the need to sample the number, times and types of state transitions as the MJP is approximated with a continuous process. Moreover, the latent variables corresponding to unobserved states can now be efficiently simulated by an Euler-Maruyama scheme which is computationally more efficient than the SSA. This approach has been followed by [13] and [12] for inference over the unknown parameters  $\boldsymbol{\theta}$  while in [27] a similar methodology has been applied on a real data from an auto-regulatory gene expression network.

### 3.2 Linear noise approximation

Substituting equation (2) in the Langevin equation (6) and dividing by  $\Omega$  we get

$$dz_t = \mathbf{S} \tilde{\mathbf{f}}(z_t, \boldsymbol{\theta}, t)dt + \frac{1}{\sqrt{\Omega}} \mathbf{S} \sqrt{\text{diag}[\tilde{\mathbf{f}}(z_t, \boldsymbol{\theta}, t)]} d\mathbf{B}_t \quad (7)$$

from which we can see that the fluctuations are of the order of  $1/\sqrt{\Omega}$  and in the thermodynamic limit (7) reduces to the Macroscopic Rate Equation (MRE)

$$\lim_{\Omega \rightarrow \infty} dz_t = \mathbf{S} \tilde{\mathbf{f}}(z_t, \boldsymbol{\theta}, t)dt.$$

To obtain the Linear Noise Approximation (LNA) we make the assumption that for sufficiently large  $\Omega$  a solution to (7) will differ from the MRE by a stochastic term of order  $1/\sqrt{\Omega}$ . That is

$$\mathbf{z}_t = \boldsymbol{\phi}_t + \frac{1}{\sqrt{\Omega}}\boldsymbol{\xi}_t \quad (8)$$

where  $\boldsymbol{\phi}_t$  are deterministic or sure variables satisfying the MRE and  $\boldsymbol{\xi}_t$  are stochastic variables. Rewriting the transition rate functions using (8) and Taylor expand around  $\boldsymbol{\phi}$  we get

$$\tilde{f}_j(\mathbf{z}, \boldsymbol{\theta}, t) = \tilde{f}_j\left(\boldsymbol{\phi} + \frac{1}{\sqrt{\Omega}}\boldsymbol{\xi}\right) = \tilde{f}_j(\boldsymbol{\phi}, \boldsymbol{\theta}, t) + \frac{1}{\sqrt{\Omega}} \sum_{d=1}^D \frac{\partial \tilde{f}_j(\boldsymbol{\phi}, \boldsymbol{\theta}, t)}{\partial \phi_i} \xi_i + O(\Omega^{-1}). \quad (9)$$

We can now substitute (8) and (9) back into (7) and collect terms of  $O(1)$  to get the expression for the differential of  $\boldsymbol{\phi}$  which is nothing other than the MRE

$$d\boldsymbol{\phi}_t = \mathbf{S}\tilde{\mathbf{f}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}, t)dt. \quad (10)$$

Finally, collecting remaining terms and neglecting terms of  $O(1/\sqrt{\Omega})$  and higher we get the differential of  $\boldsymbol{\xi}$  as

$$d\boldsymbol{\xi}_t = \mathbf{S}\mathbf{J}_{\tilde{\mathbf{f}}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}, t)\boldsymbol{\xi}_t dt + \mathbf{S}\sqrt{\text{diag}[\tilde{\mathbf{f}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}, t)]}d\mathbf{B}_t \quad (11)$$

where we used  $\mathbf{J}_{\tilde{\mathbf{f}}}(\cdot)$  to denote the Jacobian of the transition rates  $\tilde{\mathbf{f}}(\cdot)$ . Equation (11) characterises the fluctuations around the deterministic state  $\boldsymbol{\phi}$  and its validity depends on the size of  $\Omega$ . As  $\Omega$  increases the magnitude of the individual jumps  $\mathbf{s}_j$  becomes negligible relative to the distance in  $\boldsymbol{\phi}$  over which the non-linearity of  $\tilde{f}_j(\cdot)$  becomes noticeable. A measure of the sufficiency of LNA is the coefficient of variation, i.e. the ratio of the standard deviation to the mean. For a more thorough discussion on the validity of LNA the reader is referred to [23] and the supplementary material of [15].

### 3.3 Solution of the LNA and the approximate likelihood function

LNA provides a convenient expression for the approximate likelihood since the MRE (10) can be easily solved numerically and its computational cost is polynomial in  $D$ . Moreover, equation (11) is a system of linear stochastic differential equations which has an explicit solution of the form

$$\boldsymbol{\xi}_t = \boldsymbol{\Phi}(t_0, t) \left( \boldsymbol{\xi}_0 + \int_{t_0}^t \boldsymbol{\Phi}(s, t)^{-1} \mathbf{S} \sqrt{\text{diag}[\tilde{\mathbf{f}}(\boldsymbol{\phi}_s, \boldsymbol{\theta}, s)]} d\mathbf{B}_s \right) \quad (12)$$

where the integral is in the Itô sense and  $\boldsymbol{\Phi}(t_0, t)$  is the solution of

$$d\boldsymbol{\Phi}(t_0, s) = \mathbf{S}\mathbf{J}_{\tilde{\mathbf{f}}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}, t)\boldsymbol{\Phi}(t_0, s)ds, \quad \boldsymbol{\Phi}(t_0, t_0) = \mathbf{I}. \quad (13)$$

Since the Itô integral of a deterministic function is a Gaussian random variable [28], equation (12) implies that  $\boldsymbol{\xi}_t$  has a multivariate normal distribution. To simplify further the analysis assume that the initial condition for  $\mathbf{z}_t$  has a multivariate normal distribution such that  $\mathbf{z}_{t_0} \sim \mathcal{N}(\boldsymbol{\phi}_{t_0}, \mathbf{V}_{t_0})$ . For the rest of the paper we will assume that  $\boldsymbol{\phi}_{t_0}$  and  $\mathbf{V}_{t_0}$  are known. In cases where the initial conditions are unknown they can be treated as additional parameters. Equations (8, 10, 11, 12) and the specification of initial conditions further imply that

$$\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\phi}_t, \Omega^{-1}\mathbf{V}_t) \quad (14)$$

where  $\boldsymbol{\phi}_t$  are solutions of the MRE and  $\mathbf{V}_t$  are solutions of

$$d\mathbf{V}_t = \mathbf{S}\mathbf{J}_{\tilde{\mathbf{f}}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}, t)\mathbf{V}_t + \mathbf{V}_t\mathbf{J}_{\tilde{\mathbf{f}}}^T(\boldsymbol{\phi}_t, \boldsymbol{\theta}, t)\mathbf{S}^T + \mathbf{S}\text{diag}[\tilde{\mathbf{f}}(\boldsymbol{\phi}_t, \boldsymbol{\theta}, t)]\mathbf{S}^T.$$

Finally, multiplying (14) by  $\Omega$  we get

$$\mathbf{x}_t \sim \mathcal{N}(\Omega\boldsymbol{\phi}_t, \Omega\mathbf{V}_t).$$

Assume that we have observations from the stochastic process  $\mathbf{X}(t)$  at discrete time points  $t_i \in \{t_1, \dots, t_N\}$ . Moreover, assume that each observation  $\mathbf{x}_t$  is obtained by a independent realisation of  $\mathbf{X}(t)$ . For example to obtain an observation at  $t_1 = 10$  the SSA is used to simulate a trajectory from  $t_0$  to  $t_1$  and the state of the system at  $t_1$  is kept. For  $t_2 = 20$  the SSA is again used to simulate a new trajectory from  $t_0$  to  $t_2$  keeping only the state of the system at  $t_2$  and the process continues until all necessary observations are gathered. This kind of data are very frequently encountered in biology where in order to obtain a single measurement the sample has to be “sacrificed”. This is common in data obtained using Polymerase Chain Reaction reporter assays [29] for example. See also [15] for an example of an inference problem with such data. Due to the independence between different observations and the Markov property the likelihood is simply

$$p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{i=1}^N \mathcal{N}(\mathbf{x}_{t_i}|\Omega\boldsymbol{\phi}_{t_i}, \Omega\mathbf{V}_{t_i}). \quad (15)$$

In this paper we only consider observations of this kind. However the methodology is readily applicable when observations from a single realisation of  $\mathbf{X}(t)$  are available. In this case the likelihood also has a simple form

$$p(\mathbf{X}|\boldsymbol{\theta}) = \mathcal{N}[\mathbf{X}|\Omega\boldsymbol{\mu}(\boldsymbol{\theta}), \Omega\boldsymbol{\Sigma}(\boldsymbol{\theta})]$$

where  $\mathbf{X} = (\mathbf{x}_{t_1}, \dots, \mathbf{x}_{t_N})^T$  is an  $ND$  vector with all the observations,  $\boldsymbol{\mu}(\boldsymbol{\theta}) = (\boldsymbol{\phi}_{t_1}, \dots, \boldsymbol{\phi}_{t_N})^T$ , is also a  $ND$  vector with solutions of the MRE and  $\boldsymbol{\Sigma}(\boldsymbol{\theta})$  is a  $ND \times ND$  block matrix  $\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \{\boldsymbol{\Sigma}(\boldsymbol{\theta})^{i,j} : i, j \in [1, \dots, N]\}$  such that

$$\boldsymbol{\Sigma}(\boldsymbol{\theta})^{i,j} = \begin{cases} \mathbf{V}_{t_i}, & i = j \\ \mathbf{V}_{t_i}\boldsymbol{\Phi}(t_i, t_j)^T, & i \neq j \end{cases} \quad (16)$$

This stems from the fact that due to the Markov property and equation (14) each  $\mathbf{x}_{t_i}$  can be written as a sum of multivariate normal random variables and therefore  $\mathbf{X}$  is also a multivariate normal random variable. For more details refer to the supplementary material of [15] and [2]. The only additional complication which arises for time-series data is that the off-diagonal components of the LNA variance in equation (16) need to be estimated by numerically solving the system of ODEs in equation (13). Notice that despite the fact that the variance matrix is full we can still exploit the Markov property and write the likelihood as a product of the conditional likelihoods and therefore avoid the cost of inverting the  $ND \times ND$  variance matrix.

## 4 Markov Chain Monte Carlo Methods

In this section we give a brief overview of the MCMC algorithms that we consider in this work. Some familiarity with the concepts of MCMC is required by the reader since an introduction to the subject is out of the scope of this paper.

### 4.1 Metropolis-Hastings

For a random vector  $\boldsymbol{\theta} \in \mathbb{R}^D$  with density  $p(\boldsymbol{\theta})$  the Metropolis-Hastings algorithm employs a proposal mechanism  $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{t-1})$  and proposed moves are accepted with probability

$$\min \{1, p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^{t-1}|\boldsymbol{\theta}^*)/p(\boldsymbol{\theta}^{t-1})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{t-1})\}$$

. In the context of Bayesian inference the target density  $p(\boldsymbol{\theta})$  corresponds to the posterior distribution of the model parameters. Tuning the Metropolis-Hastings algorithm involves selecting the right proposal mechanism. A common choice is to use a random walk Gaussian proposal of the form  $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{t-1}) = \mathcal{N}(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{t-1}, \boldsymbol{\Sigma})$ , where  $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$  denotes the multivariate normal density with mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ .

Selecting the covariance matrix however, is far from trivial in most cases since knowledge about the target density is required. Therefore a more simplified proposal mechanism is often

considered where the covariance matrix is replaced with a diagonal matrix such as  $\Sigma = \epsilon \mathbf{I}$  where the value of the scale parameter  $\epsilon$  has to be tuned in order to achieve fast convergence and good mixing. Small values of  $\epsilon$  imply small transitions and result in high acceptance rates while the mixing of the Markov Chain is poor. Large values on the other hand, allow for large transitions but they result in most of the samples being rejected.

Tuning the scale parameter becomes even more difficult in problems where the standard deviations of the marginal posteriors differ substantially, since different scales are required for each dimension, and this is exacerbated when correlations between different variables exist. Adaptive schemes for the Metropolis-Hastings algorithm have also been proposed [30] though they should be applied with care [31]. Parameters such as reaction rate constants often differ orders of magnitude, thus a scaled diagonal covariance matrix will be a bad choice for such problems. In the numerical simulations in the next section we used a Metropolis within Gibbs scheme where each parameter is updated conditional on all others using a univariate normal density with a parameter-specific scale parameter. This allows us to tune the scale for each proposal independently and achieve better mixing.

## 4.2 Manifold Metropolis Adjusted Langevin Algorithm

Denoting the log of the target density as  $\mathcal{L}(\theta) = \log p(\theta)$ , the manifold MALA (MMALA) method, [22], defines a Langevin diffusion with stationary distribution  $p(\theta)$  on the Riemann manifold of density functions with metric tensor  $\mathbf{G}(\theta)$ . By employing a first order Euler integrator to solve the diffusion a proposal mechanism with density  $q(\theta^*|\mu(\theta^{t-1}, \epsilon), \epsilon^2 \mathbf{G}^{-1}(\theta^{t-1}))$  is obtained, where  $\epsilon$  is the integration step size, a parameter which needs to be tuned, and the  $d$ th component of the mean function  $\mu(\theta, \epsilon)_d$  is

$$\mu(\theta, \epsilon)_d = \theta_d + \frac{\epsilon^2}{2} (\mathbf{G}^{-1}(\theta) \nabla_{\theta} \mathcal{L}(\theta))_d - \epsilon^2 \sum_{i=1}^D \sum_{j=1}^D \mathbf{G}(\theta)_{i,j}^{-1} \Gamma_{i,j}^d \quad (17)$$

where  $\Gamma_{i,j}^d$  are the Christoffel symbols of the metric in local coordinates [32].

Similarly to MALA [18], due to the discretisation error introduced by the first order approximation, convergence to the stationary distribution is not guaranteed anymore and thus the Metropolis-Hastings ratio is employed to correct this bias. The MMALA algorithm can be simply stated as in Algorithm 1 and more details can be found in [22].

---

### Algorithm 1 MMALA

---

```

1: Initialise  $\theta^0$ 
2: for  $t = 1$  to  $T$  do
3:    $\theta^* \sim \mathcal{N}(\theta | \mu(\theta^{t-1}, \epsilon), \epsilon^2 \mathbf{G}^{-1}(\theta^{t-1}))$ 
4:    $r = \min \{1, p(\theta^*) q(\theta^{t-1} | \theta^*) / p(\theta^{t-1}) q(\theta^* | \theta^{t-1})\}$ 
5:    $u \sim \mathcal{U}_{[0,1]}$ 
6:   if  $r > u$  then
7:      $\theta^t = \theta^*$ 
8:   else
9:      $\theta^t = \theta^{t-1}$ 
10:  end if
11: end for

```

---

We can interpret the proposal mechanism of MMALA as a local Gaussian approximation to the target density similar to the adaptive Metropolis-Hastings of [33]. In contrast to [33], the effective covariance matrix in MMALA is the inverse of the metric tensor evaluated at the current position and no samples from the chain are required in order to estimate it, therefore avoiding the difficulties of adaptive MCMC discussed in [31]. Furthermore a simplified version of the MMALA algorithm (SMMALA) can also be derived by assuming a manifold with constant



curvature, thus cancelling the last term in Equation (17) which depends on the Christoffel symbols. Finally, the MMALA algorithm can be seen as a generalisation of the original MALA [18] since, if the metric tensor  $\mathbf{G}(\boldsymbol{\theta})$  is equal to the identity matrix corresponding to an Euclidean manifold, then the original algorithm is recovered.

### 4.3 Manifold Hamiltonian Monte Carlo

The Riemann manifold Hamiltonian Monte Carlo (RMHMC) method defines a Hamiltonian on the Riemann manifold of probability density functions by introducing the auxiliary variables  $\mathbf{p} \sim \mathcal{N}(\mathbf{0}, \mathbf{G}(\boldsymbol{\theta}))$ , which are interpreted as the momentum at a particular position  $\boldsymbol{\theta}$  and by considering the negative log of the target density as a potential function. More formally, the Hamiltonian defined on the Riemann manifold is:

$$H(\boldsymbol{\theta}, \mathbf{p}) = -\mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2} \log(2\pi|\mathbf{G}(\boldsymbol{\theta})|) + \frac{1}{2} \mathbf{p}^T \mathbf{G}(\boldsymbol{\theta})^{-1} \mathbf{p} \quad (18)$$

where the terms  $-\mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2} \log(2\pi|\mathbf{G}(\boldsymbol{\theta})|)$  and  $\frac{1}{2} \mathbf{p}^T \mathbf{G}(\boldsymbol{\theta})^{-1} \mathbf{p}$  are the potential energy and kinetic energy terms, respectively. Simulating the Hamiltonian requires a time-reversible and volume preserving numerical integrator. For this purpose the Generalised Leapfrog algorithm can be employed and provides a deterministic proposal mechanism for simulating from the conditional distribution, i.e.  $\boldsymbol{\theta}^* | \mathbf{p} \sim p(\boldsymbol{\theta}^* | \mathbf{p})$ . More details about the Generalised Leapfrog integrator can be found in [22]. To simulate a path across the manifold, the Leapfrog integrator is iterated  $L$  times which along with the integration step size  $\epsilon$  are parameters requiring tuning. Again, due to the integration errors on simulating the Hamiltonian, in order to ensure convergence to the stationary distribution the Metropolis-Hastings ratio is applied. Moreover, following the suggestion in [20] the number of Leapfrog iterations  $L$  is randomised in order to improve mixing. The RMHMC algorithm is given in Algorithm 2.

---

#### Algorithm 2 RMHMC

---

```

1: Initialise  $\boldsymbol{\theta}^0$ 
2: for  $t = 1$  to  $T$  do
3:    $\mathbf{p}_*^0 \sim \mathcal{N}(\mathbf{p} | \mathbf{0}, \mathbf{G}(\boldsymbol{\theta}^{t-1}))$ 
4:    $\boldsymbol{\theta}_*^0 = \boldsymbol{\theta}^{t-1}$ 
5:    $e \sim \mathcal{U}_{[0,1]}$ 
6:    $N = \text{ceil}(\epsilon L)$ 
   {Simulate the Hamiltonian using a generalised Leapfrog integrator for N steps}
7:   for  $n = 0$  to  $N$  do
8:     solve  $\mathbf{p}_*^{n+\frac{1}{2}} = \mathbf{p}_*^n - \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} H \left( \boldsymbol{\theta}_*^n, \mathbf{p}_*^{n+\frac{1}{2}} \right)$ 
9:     solve  $\boldsymbol{\theta}_*^{n+1} = \boldsymbol{\theta}_*^n + \frac{\epsilon}{2} \left[ \nabla_{\mathbf{p}} H \left( \boldsymbol{\theta}_*^n, \mathbf{p}_*^{n+\frac{1}{2}} \right) + \nabla_{\mathbf{p}} H \left( \boldsymbol{\theta}_*^{n+1}, \mathbf{p}_*^{n+\frac{1}{2}} \right) \right]$ 
10:     $\mathbf{p}_*^{n+1} = \mathbf{p}_*^{n+\frac{1}{2}} - \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} H \left( \boldsymbol{\theta}_*^{n+1}, \mathbf{p}_*^{n+\frac{1}{2}} \right)$ 
11:   end for
12:    $(\boldsymbol{\theta}^*, \mathbf{p}^*) = (\boldsymbol{\theta}_*^{N+1}, \mathbf{p}_*^{N+1})$ 
   {Metropolis-Hastings ratio}
13:    $r = \min \{1, \exp(-H(\boldsymbol{\theta}^*, \mathbf{p}^*) + H(\boldsymbol{\theta}^{t-1}, \mathbf{p}^{t-1}))\}$ 
14:    $u \sim \mathcal{U}_{[0,1]}$ 
15:   if  $r > u$  then
16:      $\boldsymbol{\theta}^t = \boldsymbol{\theta}^*$ 
17:   else
18:      $\boldsymbol{\theta}^t = \boldsymbol{\theta}^{t-1}$ 
19:   end if
20: end for

```

---

Similar to the MMALA algorithm, when the metric tensor  $\mathbf{G}(\boldsymbol{\theta})$  is equal to the identity matrix corresponding to an Euclidean manifold, then RMHMC is equivalent to the HMC algorithm of [19].

## 5 Implementation details

### 5.1 Gradient and metric tensor for the LNA

For the manifold MCMC algorithms discussed in this section we will need the gradient of the log likelihood as well as a metric tensor for the LNA. For density functions the natural metric tensor is the expected Fisher Information,  $\mathbf{I}(\boldsymbol{\theta})$ , [34] and for a multivariate normal with mean  $\boldsymbol{\mu}(\boldsymbol{\theta})$  and covariance matrix  $\boldsymbol{\Sigma}(\boldsymbol{\theta})$  its general form is

$$I(\boldsymbol{\theta})_{i,j} = \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_i} \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}) \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_j} + \frac{1}{2} \text{Tr} \left( \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}) \frac{\partial \boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial \theta_i} \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}) \frac{\partial \boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial \theta_j} \right).$$

For the likelihood in equation (15) the Fisher Information is then a sum of  $N$  matrices  $\mathbf{I}(\boldsymbol{\theta}, t)$ , one evaluated at each time point. Similarly the general form of the partial derivatives for the log of a multivariate normal is

$$\frac{\partial \ln \mathcal{N}[\mathbf{x} | \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\Sigma}(\boldsymbol{\theta})]}{\partial \theta_i} = \frac{1}{2} \text{Tr} \left[ (\mathbf{c} \mathbf{c}^T - \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})) \frac{\partial \boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial \theta_i} \right] + \mathbf{c}^T \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_i}$$

where  $\mathbf{c} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})[\mathbf{x} - \boldsymbol{\mu}(\boldsymbol{\theta})]$ .

Moreover, during the leap-frog integration for the RMHMC and for the mean function of MMALA the partial derivatives of the Fisher Information are needed. Their general form is

$$\begin{aligned} \frac{\partial I(\boldsymbol{\theta})_{i,j}}{\partial \theta_k} &= \frac{\partial^2 \boldsymbol{\mu}(\boldsymbol{\theta})^T}{\partial \theta_i \partial \theta_k} \mathbf{a}_j + \mathbf{a}_i^T \frac{\partial^2 \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_j \partial \theta_k} - \mathbf{a}_i^T \frac{\partial \boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial \theta_k} \mathbf{a}_j \\ &\quad - \frac{1}{2} \text{Tr} [\mathbf{A}_k (\mathbf{A}_i \mathbf{A}_j + \mathbf{A}_j \mathbf{A}_i)] \\ &\quad + \frac{1}{2} \text{Tr} \left[ \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}) \left( \frac{\partial \boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_k} \mathbf{A}_j + \frac{\partial \boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial \theta_j \partial \theta_k} \mathbf{A}_i \right) \right] \end{aligned}$$

where  $\mathbf{a}_i = \boldsymbol{\Sigma}^{-1} \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_i}$  and  $\mathbf{A}_i = \boldsymbol{\Sigma}^{-1} \frac{\partial \boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial \theta_i}$ .

The above quantities require first and second order sensitivities for the  $\boldsymbol{\phi}$  and  $\mathbf{V}$  which we obtain by augmenting the ODE systems with the additional sensitivity equations. For an ODE system of  $n_y$  equations with form  $\dot{\mathbf{y}} = \mathbf{F}(\mathbf{y}, t, \boldsymbol{\theta})$ ,  $\mathbf{y}(t_0) = \mathbf{y}_0(\boldsymbol{\theta})$  and  $n_\theta$  parameters  $\boldsymbol{\theta}$ , the first and second order forward sensitivity equations are given by (19) and (20) respectively.

$$\frac{\partial \dot{\mathbf{y}}}{\partial \boldsymbol{\theta}} = \mathbf{F}_{\mathbf{y}} \frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}} + \mathbf{F}_{\boldsymbol{\theta}}, \quad \frac{\partial \mathbf{y}(t_0)}{\partial \boldsymbol{\theta}} = \frac{\partial \mathbf{y}_0}{\partial \boldsymbol{\theta}} \quad (19)$$

$$\begin{aligned} \frac{\partial^2 \dot{\mathbf{y}}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} &= [\mathbf{F}_{\mathbf{y}} \otimes \mathbf{I}_{n_\theta}] \frac{\partial^2 \mathbf{y}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} + \left[ \mathbf{I}_{n_y} \otimes \frac{\partial \mathbf{y}^T}{\partial \boldsymbol{\theta}} \right] \left[ \mathbf{F}_{\mathbf{y}, \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}} + \mathbf{F}_{\mathbf{y}, \boldsymbol{\theta}} \right] \\ &\quad + \left[ \mathbf{F}_{\boldsymbol{\theta}, \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}} + \mathbf{F}_{\boldsymbol{\theta}, \boldsymbol{\theta}} \right], \\ \frac{\partial^2 \mathbf{y}(t_0)}{\partial \boldsymbol{\theta}^2} &= \frac{\partial^2 \mathbf{y}_0}{\partial \boldsymbol{\theta}^2} \end{aligned} \quad (20)$$

We use  $\mathbf{F}_{\boldsymbol{\theta}}$  to denote the  $n_y \times n_\theta$  matrix where its  $j^{th}$  column is the partial derivatives of  $\mathbf{F}$  with respect to  $\theta_j$ .  $\mathbf{F}_{\boldsymbol{\theta}, \mathbf{y}}$  denotes the derivative of  $\mathbf{F}_{\boldsymbol{\theta}}$  with respect to  $\mathbf{y}$  and is an  $n_\theta \cdot n_y \times n_y$  matrix where its  $j^{th}$  column is the partial derivatives of  $\text{vec}(\mathbf{F}_{\boldsymbol{\theta}}^T)$  with respect to  $y_j$ .  $\mathbf{I}_{n_y}$  denotes the  $n_y \times n_y$  Identity matrix,  $\otimes$  the Kronecker product and  $\text{vec}(\mathbf{A})$  an operator that creates a column vector by stacking the columns of matrix  $\mathbf{A}$ .

## 5.2 Re-parameterisation

In many problems the parameters  $\theta$  can be constrained in certain parts of  $\mathbb{R}^{n_\theta}$  where  $n_\theta$  is the number of parameters. In models of chemical kinetics for example, rate parameters must be positive and can differ by orders of magnitude. For the MCMC algorithms described in the previous section we will need a re-parameterisation in order to allow the algorithms to operate on an unbounded and unconstrained parameter space.

For the numerical simulations in section 6 we use a  $\log_{10}$  re-parameterisation by introducing the variables  $\check{\theta}_p = \log_{10}(\theta_p)$ ,  $p \in [1, \dots, n_\theta]$ . To ensure that we sample from the correct posterior the joint density is scaled by the determinant of the Jacobian such that  $p(\mathbf{X}|\check{\theta})p(\check{\theta})|\mathbf{J}(\check{\theta})|$  where  $\mathbf{J}(\check{\theta})$  is a  $n_\theta \times n_\theta$  diagonal matrix with elements  $\mathbf{J}(\check{\theta})_{p,p} = 10^{\check{\theta}_p} \log(10)$ .

The gradient and Fisher information along with its partial derivatives follow from the chain rule as

$$\begin{aligned}\nabla_{\check{\theta}} \mathcal{L}(\check{\theta}) &= \nabla_{\theta} \mathcal{L}(\theta) \mathbf{J}(\check{\theta}) \\ \mathbf{I}(\check{\theta}) &= \mathbf{J}(\check{\theta})^T \mathbf{I}(\theta) \mathbf{J}(\check{\theta}) \\ \frac{\partial \mathbf{I}(\check{\theta})}{\partial \check{\theta}_p} &= 2\mathbf{J}(\check{\theta})^T \mathbf{I}(\theta) \frac{\partial \mathbf{J}(\check{\theta})}{\partial \check{\theta}_p} + \mathbf{J}(\check{\theta})^T \frac{\partial \mathbf{I}(\theta)}{\partial \theta_p} \mathbf{J}(\check{\theta}) \frac{\partial \theta_p}{\partial \check{\theta}_p}\end{aligned}$$

## 5.3 Choice of priors

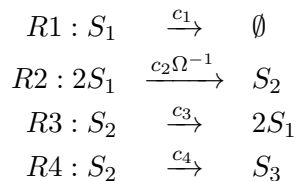
In Bayesian statistics priors provide the means for incorporating existing knowledge for the parameters in question. The choice of a suitable prior distribution can be informed from knowledge about the process being modelled, the experimental design and empirical observations. For example we might want to restrict rate parameters in chemical kinetics from becoming very high since we assume from the experimental design that reactions are slow enough to be able to be observed. In some cases the model itself can also guide the choice of the prior. For example when a model is only defined for a certain range of values of the parameters, a prior restricting the parameters in that range should be used.

In the numerical simulations of the next section we use independent normal priors for the parameters  $\check{\theta}$ . Due to the re-parameterisation introduced earlier, this corresponds to a log-normal prior with base 10 for the parameters  $\theta$ . This choice allows parameters to differ several orders of magnitude while it ensures they are strictly positive. Moreover, as noted in [22] the negative Hessian of the prior is added to the Fisher information in order to form the metric tensor used during MCMC sampling. This has the added benefit of regularising the Fisher information when it is near-singular [9] although we have not observed such problems in the simulations presented here.

# 6 Numerical Simulations

## 6.1 Chemical kinetics

In this section we consider two examples from chemical kinetics [14] and study the effect of the system size parameter on inference using MCMC. The first system consists of three species where an unstable monomer,  $S_1$ , can dimerise to an unstable dimer,  $S_2$ , which is then converted to a stable form,  $S_3$ . The reaction set for this system is



min. ESS vs. $\Omega$			
$\Omega$	M.H.	SMMALA	RMHMC
1	121 (3.6)	150 (3.9)	245 (0.06)
2	226 (6.7)	2163 (57.2)	4775 (1.3)
5	132 (3.9)	3539 (93.6)	4618 (1.2)
10	180 (5.3)	3397 (89.8)	5954 (1.6)
100	214 (6.4)	3725 (98.5)	6066 (1.7)

Table 1: Comparison of minimum Effective Sample Size (ESS) and time normalised min. ESS for different values of the system size parameter  $\Omega$  of the decay dimerisation reaction model. Time normalised ESS is given in parenthesis. Results are calculated from 10,000 posterior samples.

and the state of the system at time  $t$  will be denoted by  $\mathbf{X}(t) = [S_1(t), S_2(t), S_3(t)]^T$ . The propensity functions, or state transition probabilities are  $\mathbf{f}(\mathbf{X}, \boldsymbol{\theta}) = [c_1 S_1(t), c_2 \Omega^{-1} S_1(t)(S_1(t) - 1)/2, c_3 S_2(t), c_4 S_3(t)]^T$  and the corresponding state change matrix is

$$\mathbf{S} = \begin{pmatrix} -1 & -2 & 2 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (21)$$

For our experiments we will assume that initial conditions are known and set them to  $S_1(t_0) = 5\Omega$ ,  $S_2(t_0) = S_3(t_0) = 0$ ,  $t_0 = 0$ . Moreover we will set the reaction rate parameters to  $c_1 = 1$ ,  $\hat{c}_2 = 2\Omega^{-1}$ ,  $c_3 = 0.5$  and  $c_4 = 0.04$ . Notice that we make explicit the relation between the system size and parameter  $\hat{c}_2$  and we will infer rate  $c_2$  up to a proportionality constant. For all the experiments we simulate data using the SSA of [1] for the time interval  $t \in [0, 10]$  and we discretise such that  $t_i - t_{i-1} = 0.1$ . Each observation  $\mathbf{X}(t_i)$  is obtained independently by simulating a trajectory from  $t_0$  to  $t_i$  and keeping only the last state discarding the rest of the trajectory. Moreover for each time point  $t_i$  we also simulate 10 independent observations. Since each observation is obtained by a different trajectory of the MJP we assume that initial conditions do not have a point mass rather for each trajectory we sample its initial condition from a Poisson with means  $S_1(0), S_2(0), S_3(0)$ .

We use the synthetic data to perform inference for the rate parameters  $\boldsymbol{\theta} = (c_1, \hat{c}_2, c_3, c_4)^T$  by drawing samples from the posterior

$$p(\boldsymbol{\theta}|\mathbf{X}) \propto p(\boldsymbol{\theta}) \prod_{i=1}^N \prod_{r=1}^{10} \mathcal{N}[\mathbf{X}_r(t_i) | \Omega \boldsymbol{\phi}(t_i), \Omega \mathbf{V}(t_i)]$$

where  $r$  indexes independent observations for the same time point. For all simulations in this paper we assume that the means for the initial conditions are known. Following similar arguments as for the derivation of the LNA in Section 3, namely that as the system approaches its thermodynamic limit transition densities become Gaussian, the initial conditions for the ODE systems for the mean and variance of the transition densities are  $\boldsymbol{\phi}(0) = \mathbf{X}(0)\Omega^{-1}$  and  $\mathbf{V}(0) = \mathbf{I}$ , where  $\mathbf{I}$  is the identity matrix. In a more realistic scenario the initial conditions must be included as additional parameters in  $\boldsymbol{\theta}$ . For all parameters we used an independent log-normal prior with base 10, zero mean and one standard deviation and chains are initialised by drawing a random sample from the prior. For the Metropolis-Hastings sampler we set the initial proposal scale parameters to  $\approx 1e^{-6}$  and automatically adapt them every 100 samples during the burn-in phase in order to achieve an acceptance rate of 25% – 30% [17]. The same adaptation strategy was followed for the simplified MMALA and RMHMC algorithms where the initial step size was also set to  $\approx 1e^{-6}$  and was tuned in order to achieve acceptance rates in the order of 70 – 80% [22]. Finally, the number of leap-frog steps for RMHMC was fixed to 5. We have found that a burn in period of 10,000 to 20,000 samples was adequate for all algorithms to converge to the stationary distribution.

Posterior mean and SD. vs. $\Omega$				
$\Omega$	$c_1$	$\hat{c}_2$	$c_3$	$c_4$
True	1	$2\Omega^{-1}$	0.5	0.04
1	0.88 (0.031)	1.72 (0.253)	0.39 (0.039)	0.003 (0.002)
2	1.3 (0.041)	0.69 (0.066)	0.35 (0.016)	0.014 (0.002)
5	0.93 (0.019)	0.39 (0.028)	0.48 (0.025)	0.034 (0.002)
10	1.0 (0.015)	0.18 (0.008)	0.47 (0.015)	0.037 (0.001)
100	0.99 (0.004)	0.01 (0.0002)	0.52 (0.004)	0.039 (0.0003)

Table 2: Marginal posterior means and standard deviations calculated from the RMHMC chain for different values of the system size parameter  $\Omega$  of the decay-dimerisation reaction model. Notice that  $\hat{c}_2$  parameter is proportional to  $\Omega$ . Results are calculated from 10,000 posterior samples.

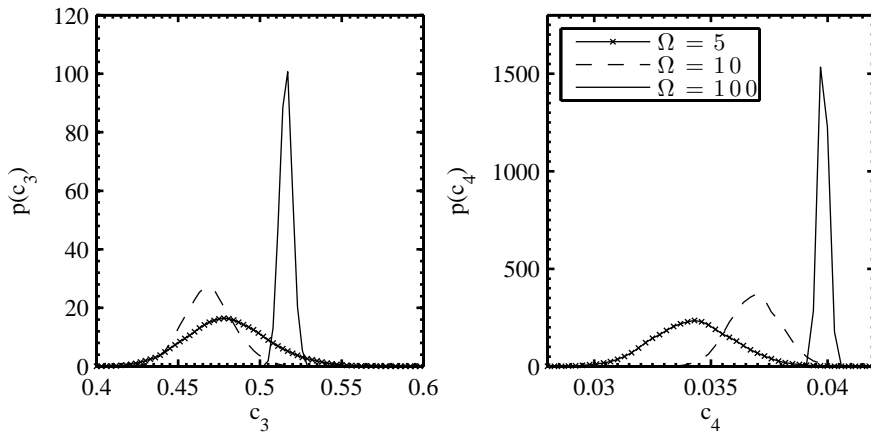


Figure 1: Marginal posteriors for parameters  $c_3$  (left panel) and  $c_4$  right panel for different values of  $\Omega$ . Results are obtained by 10,000 posterior samples using RMHMC.

Table 1 compares the minimum Effective Sample Size (ESS) and the time normalised ESS obtained by all algorithms for different values of the system size parameter  $\Omega$ . The SMMALA and RMHMC samplers utilise the gradients and the Fisher Information of the approximate likelihood obtained by the LNA in order to make efficient proposals. As the system size increases and thus the LNA better approximates the true likelihood then mixing of the manifold MCMC algorithms improves. For this particular example we can see that good mixing can be achieved even for very small systems with only  $\approx 25$  molecules, ( $\Omega = 5$ ). The M.H. sampler is not affected by the system size but its mixing is very poor in all cases. From the time normalised ESS we can also see that despite the improved mixing of RMHMC the computational cost is significant. On the contrary SMMALA provides a good tradeoff between mixing efficiency and computational cost. Finally, Table 2 reports the marginal posterior means and standard deviations for different values of  $\Omega$  obtained by RMHMC. The marginal posteriors for parameters  $c_3$  and  $c_4$  with  $\Omega \geq 5$  are also shown in Figure 1. Results from the MH and SMMALA samplers are similar and are omitted. For small system sizes we can observe that there is an increased bias of the Monte Carlo estimate while the posterior standard deviation is higher reflecting the high degree of uncertainty around the mean. The bias however significantly reduces as the system size increases and for  $\Omega \geq 5$  reasonable estimates can be obtained.

The second example from the chemical kinetics literature that we consider is the Schlögl

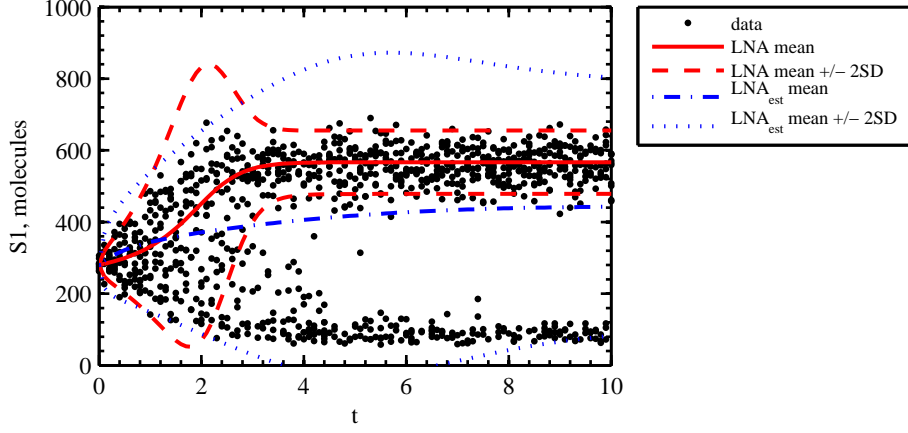
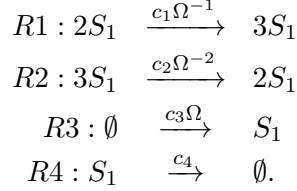


Figure 2: Simulated time point data using SSA for the Schlögl reaction set and LNA predictions. Dots correspond to simulated data. The bold and dashed red lines correspond to the LNA prediction for the means and standard deviations using the true parameters. Dotted blue lines correspond the LNA predictions using the posterior means for the rate parameters. (Online version in colour.)

$$\mathbf{S} = \begin{pmatrix} 1, & -1, & 1, & -1, \end{pmatrix} \quad (22) \quad \mathbf{f}(\mathbf{X}, \boldsymbol{\theta}) = \begin{pmatrix} c_1 \Omega^{-1} \frac{1}{2} S_1 (S_1 - 1), \\ c_2 \Omega^{-2} \frac{1}{6} S_1 (S_1 - 1) (S_1 - 2), \\ c_3 \Omega, \\ c_4 S_1 \end{pmatrix} \quad (23)$$

reaction set.



The corresponding state transition rates and state change matrix are given in equations (22) and (23) respectively. The state of the system consists only of the number of molecules of a single species  $\mathbf{X}(t) = S_1(t)$ . The system is known to have two stable states which appear at different times depending on the size of the system. [14] have shown that the LNA fails to provide a reasonable approximation of this system even for large concentration numbers. Their numerical experiments demonstrate that the LNA approximation can only approximate one of the two modes depending on the initial conditions. Here our aim is to show that using the LNA to obtain an approximate posterior over the unknown reaction rate constants can be very misleading for bi-stable systems. Using the resulting posterior means for the reaction rates gives us an LNA that fails to approximate any of the two stable modes.

To demonstrate that we follow the same experimental procedure as in the previous example. That is, we simulate data using the SSA for the time interval  $t_i \in [0, 10]$ ,  $t_i - t_{i-1} = 0.1$  with fixed rate parameters and then use this data for posterior inference of the rate parameters using MCMC. Values for the true rate parameters and initial conditions were set as in [14]. Namely,  $c_1 = 0.003$ ,  $c_2 = 0.0001$ ,  $c_3 = 200$ ,  $c_4 = 3.5$  and  $\mathbf{X}(t_0) = 280\Omega$ , where  $\Omega$  was fixed to 1. After 10,000 burn-in samples all samplers converged to a posterior distributions with mean

$$E_{p(\boldsymbol{\theta}|\mathbf{X})}[\boldsymbol{\theta}] \approx (0.130, 3.3e^{-4}, 3.5e^{+3}, 26.22)^T$$

and variance

$$\text{var}_{p(\boldsymbol{\theta}|\mathbf{X})}[\boldsymbol{\theta}] \approx (1.2e^{-4}, 8.2e^{-10}, 8.6e^{+4}, 4.53)^T$$

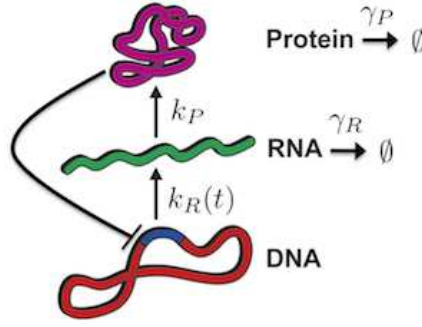
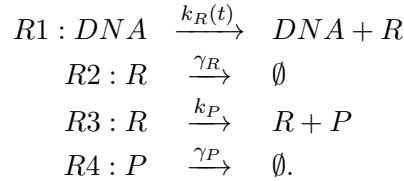


Figure 3: Schematic representation of the auto-regulatory gene expression model with a negative feedback loop. A gene is transcribed into mRNA which is translated to a protein that suppresses gene transcription. (Online version in colour.)

The LNA obtained by using the posterior means for the rate constants is shown in Figure 2 along with the data obtained by the SSA and the LNA using the true values for the rate constants. We can see that the LNA approximation obtained by the posterior means fails to approximate any of the two modes. Rather it approximates the empirical mean and variance of the data.

## 6.2 Single gene expression

Finally, to illustrate the applicability of the methodology to systems biology we also consider a simplified model for the biochemical reactions involved in the expression of a single gene to protein. The model presented in this section is the same with the model used in the study of [15] and we adopt the same notation in order to make comparisons easier. Gene expression is modelled in terms of three biochemical species; DNA, mRNA and protein; and four chemical reactions or state transitions; transcription, mRNA degradation, translation and protein degradation. The model can be written in chemical reaction notation as



The system state at time  $t$  is  $\mathbf{X}(t) = [R(t), P(t)]^T$  where  $R(t)$  and  $P(t)$  are the number of mRNA and protein molecules respectively. The corresponding state dependent transition rates are  $\mathbf{f}(\mathbf{X}, t) = [k_R(t), \gamma_R R(t), k_P R(t), \gamma_P P(t)]^T$  where  $\gamma_R, k_P$  and  $\gamma_P$  are unknown reaction rate constants.  $k_R(t)$  is the time dependent transcription rate of the gene which for the purposes of this section is modelled as

$$k_R(t) = b_0 \exp(-b_1(t - b_2)^2) + b_3$$

where all the  $b_i$ s are also unknown parameters controlling gene transcription. This corresponds to a transcription rate that due to some stimulus (experimental or environmental) increases for  $t < b_2$  and then it drops towards the base line  $b_3$  for  $t > b_2$ . Finally, the state change matrix for this set of reactions is given in equation (24).

$$\mathbf{S} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}. \quad (24)$$

As in the study of [15] we also consider a non-linear extension of this model where the transcription rate of the gene  $k_R(t)$  is a function of the protein concentration that the gene is transcribed to. This is modelled using a Hill function

$$\hat{k}_R(t, P) = k_R(t) / (1 + (P/H)^{n_H})$$

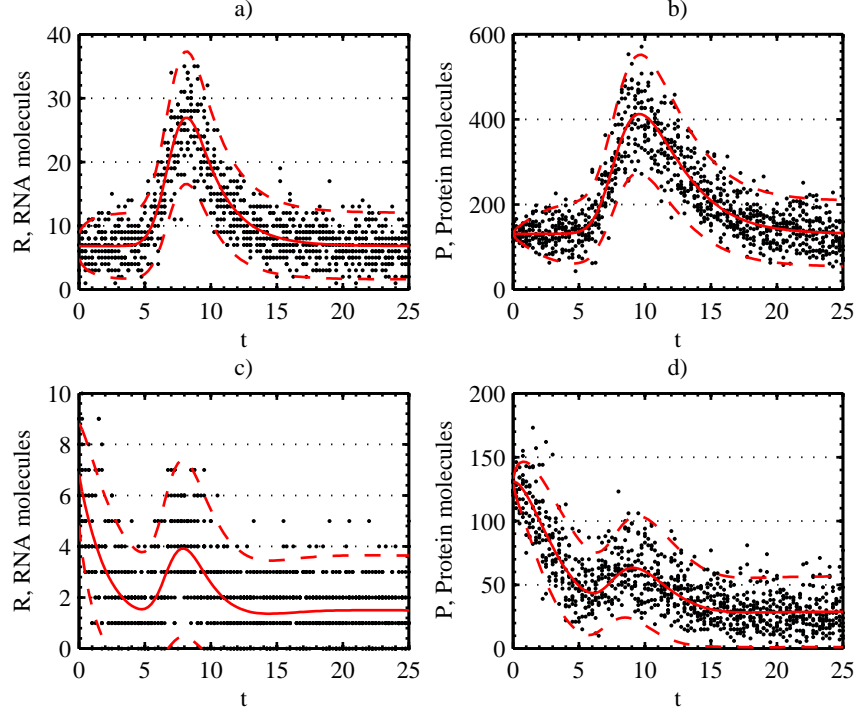


Figure 4: Data simulated from the single gene expression model using SSA. Figures (a), (b), for the linear model and Figures (c), (d), for the auto-regulatory model. Dots correspond to 10 independent draws for each time point. The bold line is the mean predicted by LNA with the true model parameters and the dashed lines are the  $\pm 2\times$  standard deviation predicted by LNA. Left column shows the mRNA molecules and right column the protein. (Online version in colour.)

where for the experiments of this section we will set  $H = b_3 k_P / (2\gamma_R \gamma_P)$  and  $n_H = 1/2$  making the protein an inhibitor of mRNA transcription. A schematic representation of this model is shown in Figure 3. For the rest of this section we will refer to this model as the auto-regulatory single gene expression model.

Using the transition probabilities  $\mathbf{f}(\mathbf{X}, t)$  and matrix  $\mathbf{S}$  we simulate synthetic data using the Stochastic Simulation Algorithm (SSA) [1] and sample at discrete time points. Values for the unknown rate constants and the parameters controlling gene transcription are shown in Table 3. The time interval is taken to be  $t_i \in [0, 25]$  while the interval between two observations  $t_i - t_{i-1} = 0.25$ . Each time point is sampled from an independent trajectory by starting the SSA from  $t_0$  and simulate up to  $t_i$  keeping only the state  $\mathbf{X}(t_i)$  and discarding the rest of the trajectory. This resembles the experimental conditions often encountered in biology where in order to make an observation the sample has to be “sacrificed”. Finally for each time point we also generate 10 independent observations from different trajectories. Initial conditions  $\mathbf{X}(t_0)$  are simulated from a Poisson distribution with means  $b_3/\gamma_R$  and  $b_3 k_P / (\gamma_R \gamma_P)$  for the mRNA and protein molecules respectively. The system size parameter  $\Omega$  is considered to be unknown and for this experiment is set to 1 such that concentrations are equal to the number of molecules. Figures (4.a) and (4.b) show data simulated from this process from the single gene expression model as well as the LNA prediction. Simulated data for the auto-regulatory model are presented in Figures (4.c) and (4.d).

We use the simulated data to infer the unknown parameters  $\boldsymbol{\theta} = (\gamma_R, k_P, \gamma_P, b_0, b_1, b_2, b_5)^T$  by sampling using MCMC from the LNA approximate posterior

$$p(\boldsymbol{\theta}|\mathbf{X}) \propto p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{i=1}^N \prod_{r=1}^R \mathcal{N}[\mathbf{X}_r(t_i)|\boldsymbol{\phi}(t_i), \mathbf{V}(t_i)]$$



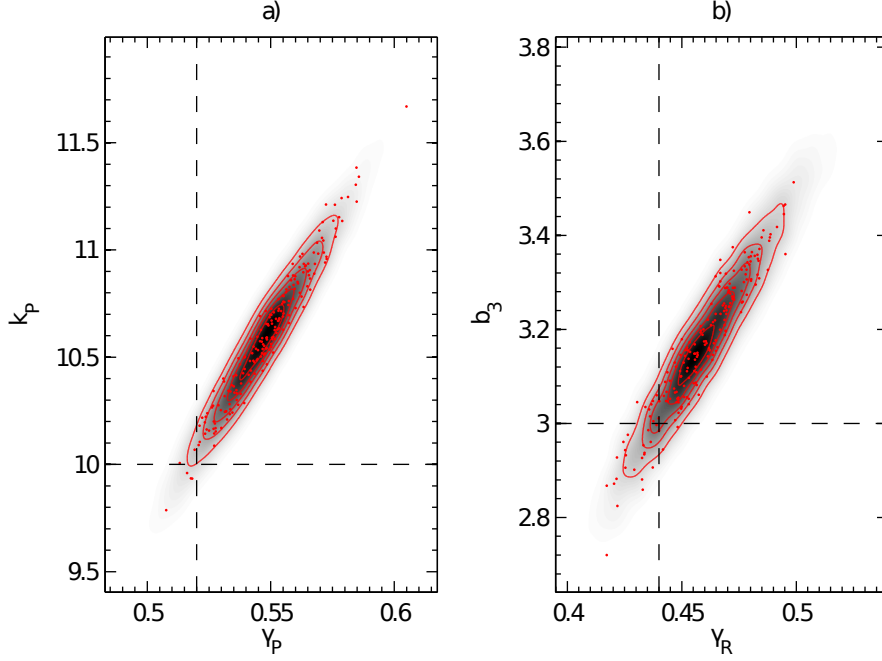


Figure 5: Marginal joint posterior for parameters  $\gamma_P, k_P$  left panel, Figure (a), and  $\gamma_R, b_3$  right panel, Figure (b) for the single gene expression model. Dashed lines are the true values used to generate the synthetic data. Dots are samples from the posterior. Iso-contours and shaded region are obtained by kernel density estimation using posterior samples. (Online version in colour.)

where  $r$  indexes independent samples for the same time point and  $R = 10$ .

Table 3 summarises the results from the MCMC chains for the two models of gene expression. Firstly, we can see that despite the relatively small number of molecules in both systems the LNA approximation provides very accurate estimates for the true parameters. Moreover we can see that the mixing of the Metropolis-Hastings sampler is very poor for both models while RMHMC and simplified Manifold MALA algorithms perform very well. This can be explained by the strong correlations between parameters in the posterior distribution preventing the M.H. sampler to make sufficiently large proposals. For example, the parameters  $k_P, \gamma_P$  control mRNA translation and protein degradation respectively. The concentration of protein molecules is directly affected by the two rates and they are expected to be heavily correlated. In Figures 5.a 5.b we show the marginal joint posterior for parameters  $k_P, \gamma_P$  and  $\gamma_R, b_3$  for the single gene expression model which exhibit very strong positive correlation. Finally figure 6 compares the trace plots obtained from MH, SMMALA and RMHMC for parameters  $\gamma_P$  and  $k_P$  of the auto-regulatory gene expression model.

## 7 Conclusions and Future Work

Bayesian inference for Markov jump processes is a challenging problem which has many important practical applications. Previous research [6] has shown that although exact inference is possible, the computational cost and the autocorrelation of the Markov chains is such that limits its applicability to small problems. The main problem stems from the requirement to simulate the MJP for the trajectory of the system between discrete observations. [13] has shown that by considering a diffusion approximation the simulation can be performed in a much more efficient manner. In this paper we considered the linear noise approximation which only requires to simulate a system of ordinary differential equations while the stochastic fluctuations have an exact analytic solution. The linear noise approximation is valid only when the system is sufficiently close to its thermodynamic limit, a condition that is also required for the diffusion

Single gene expression model.							
Parameters	$\gamma_R$	$\gamma_P$	$k_P$	$b_0$	$b_1$	$b_2$	$b_3$
True values	0.44	0.52	10.0	15.0	0.40	7.0	3.0
Metropolis-Hastings							
(A.R.)	(0.28)	(0.33)	(0.30)	(0.34)	(0.29)	(0.29)	(0.34)
( $\epsilon$ )	(0.013)	(0.007)	(0.008)	(0.022)	(0.056)	(0.007)	(0.016)
Mean	0.45	0.54	10.54	14.86	0.39	7.03	3.14
S.D.	0.017	0.017	0.336	0.509	0.029	0.056	0.149
ESS	42	34	34	149	117	58	44
ESS/time	1.42	1.15	1.15	5.05	3.96	1.96	1.49
Simplified Manifold MALA (A.R.= 0.79, $\epsilon$ = 1.05 )							
Mean	0.45	0.54	10.57	14.88	0.39	7.04	3.17
S.D.	0.018	0.016	0.306	0.537	0.030	0.053	0.152
ESS	2891	2911	2958	2787	3310	3183	2878
ESS/time	83.79	84.37	85.73	80.78	95.94	92.26	83.42
Manifold HMC (A.R.= 0.84, $\epsilon$ = 0.91, L=5 )							
Mean	0.46	0.54	10.57	14.95	0.39	7.04	3.18
S.D.	0.018	0.015	0.300	0.555	0.030	0.052	0.153
ESS	7731	8238	8304	7160	7380	7791	7950
ESS/time	0.52	0.55	0.56	0.48	0.49	0.52	0.53
Auto-regulatory single gene expression model.							
Metropolis-Hastings							
(A.R.)	(0.26)	(0.36)	(0.31)	(0.33)	(0.24)	(0.30)	(0.35)
( $\epsilon$ )	(0.028)	(0.012)	(0.016)	(0.071)	(0.231)	(0.019)	(0.029)
Mean	0.4360	0.52	10.40	14.61	0.40	6.82	3.13
S.D.	0.016	0.018	0.424	1.089	0.076	0.090	0.142
ESS	201	71	73	465	339	420	239
ESS/time	6.12	2.16	2.22	14.17	10.33	12.80	7.28
Simplified Manifold MALA (A.R.= 0.71, $\epsilon$ = 1.17)							
Mean	0.43	0.52	10.44	14.24	0.38	6.82	3.12
S.D.	0.016	0.018	0.422	1.125	0.075	0.091	0.142
ESS	2990	3270	3454	3124	3164	3316	3195
ESS/time	76.86	84.06	88.79	80.30	81.33	85.24	82.13
Manifold HMC (A.R.= 0.82, $\epsilon$ = 0.91, L=5 )							
Mean	0.43	0.52	10.43	14.52	0.40	6.82	3.13
S.D.	0.016	0.017	0.412	1.158	0.078	0.089	0.144
ESS	6532	6593	6614	5112	5384	6595	6642
ESS/time	0.41	0.41	0.41	0.32	0.34	0.41	0.42

Table 3: Marginal posterior means and standard deviations for the parameters of the single gene expression model using simulated data. The ESS is calculated for chains of 10,000 samples after a burn-in period of 10,000 iterations with initial parameters randomly sampled from the prior. Average acceptance rate (A.R.) and sampler parameters are shown in parenthesis. Notice that for the Metropolis-Hastings sampler a different proposal is used for each parameter. The prior for all parameters was  $\log_{10} \mathcal{N}(0.0, 2.0)$ .

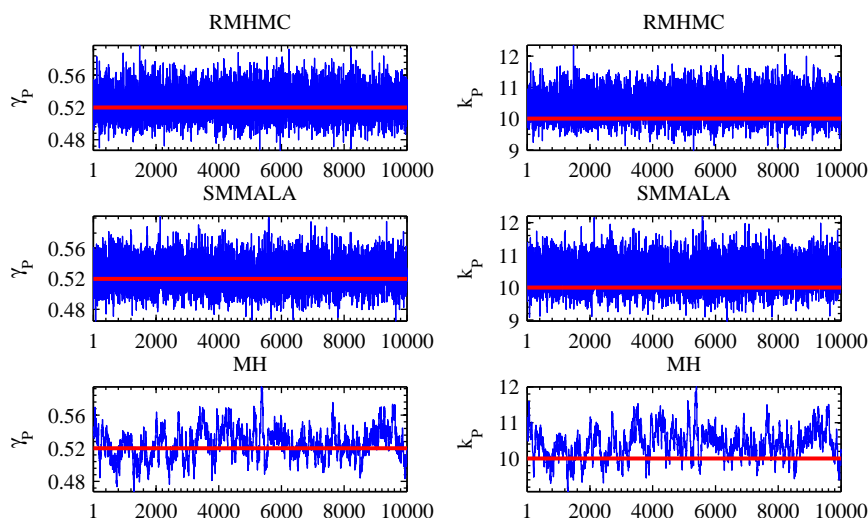


Figure 6: Example trace plots from the auto-regulatory gene expression model for parameters  $\gamma_P$  and  $k_P$ . Red solid line denotes the true values. (Online version in colour.)

approximation. Previous research on the linear noise approximation [15] has focussed on the Metropolis-Hastings sampler. We have demonstrated here that when the posterior distribution exhibits strong correlation between parameters then the Metropolis-Hastings sampler has strong auto-correlations. Such correlations are very common for chemical reaction and gene regulatory systems. The Riemann manifold MCMC algorithms we considered in this work exploit the geometric structure of the target posterior in order to design efficient proposal mechanisms. In particular the simplified Manifold MALA algorithm is a conceptually simple algorithm which provides a good trade-off between computational cost and sample auto-correlation.

Although the problems considered in this work are relatively small, but certainly non-trivial, we believe that the proposed methodology is applicable for larger and more complex systems. The systems we studied in this paper all have a linear dependence on the unknown parameters and we have not observed any local modes in our simulations. The analysis of such systems is the subject of on-going work. Moreover, in real applications it is not possible to observe the populations of all species and there is an additional measurement error term. Extension of the LNA to handle such cases is straight forward, see [15] for example, however the effect of partial observations and measurement error on the MCMC inference is something that needs to be studied in more detail.

## References

- [1] 1. Gillespie, D. T. 2007 Stochastic Simulation of Chemical Kinetics. *Annual Review of Physical Chemistry* **58** (1), 35–55, (DOI 10.1146/annurev.physchem.58.032806.104637.).
- [2] 2. Komorowski, M., Costa, M. J., Rand, D. A., and Stumpf, M. P. H. 2011 Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proceedings of the National Academy of Sciences* **108** (21), 8645–8650, (DOI 10.1073/pnas.1015814108.).
- [3] 3. Spencer, M. and Susko, E. 2005 Continuous-Time Markov Models for Species Interactions. *Ecology* **86** (12), 3272–3278, (DOI 10.1890/05-0029.).
- [4] 4. Adas, A. 1997 Traffic models in broadband networks. *Communications Magazine, IEEE* **35** (7), 82–89, (DOI 10.1109/35.601746.).
- [5] 5. Gillespie, D. T. 2005, A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications* **188** (1-3), 404–425, (DOI 10.1016/0378-4371(92)90283-V.).

- [6] 6. Boys, R. J., Wilkinson, D. J., and Kirkwood, T. B. 2008 Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing* **18**, 125–135, (DOI 10.1007/s11222-007-9043-x.).
- [7] 7. Van Kampen, N. G. 1992 *Stochastic Processes in Physics and Chemistry*, 3rd ed., Netherlands: North-Holland.
- [8] 8. Xu, T., Vyshemirsky, V., Gormand, A., von Kriegsheim, A., Girolami, M., Baillie, G. S., Ketley, D., Dunlop, A. J., Milligan, G., Houslay, M. D., and Kolch, W. 2010 Inferring signaling pathway topologies from multiple perturbation measurements of specific biochemical species. *Science Signaling* **3** (113), ra20, (DOI 10.1126/scisignal.2000517.).
- [9] 9. Calderhead, B. and Girolami, M. 2011 Statistical analysis of nonlinear dynamical systems using differential geometric sampling methods. *Interface Focus* **1** (6), 821–835, (DOI 10.1098/rsfs.2011.0051.).
- [10] 10. van Kampen, N. G. 1982 The Diffusion Approximation for Markov Processes. In *Thermodynamics & kinetics of biological processes* (ed. Lamprecht, I. and Zotin, A. I.) New York, USA: Walter de Gruyter & Co..
- [11] 11. Gillespie, D. T. 2000 The chemical Langevin equation. *The Journal of Chemical Physics* **113** (1), 297–306, (DOI 10.1063/1.481811.).
- [12] 12. Roberts, G. O. and Stramer, O. 2001 On Inference for Partially Observed Nonlinear Diffusion Models Using the Metropolis-Hastings Algorithm. *Biometrika* **88** (3), 603–621, (DOI 10.1093/biomet/88.3.603.).
- [13] 13. Golightly, A. and Wilkinson, D. J. 2011 Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus* **1** (6), 807–820, (DOI 10.1098/rsfs.2011.0047.).
- [14] 14. Wallace, E., Gillespie, D., Sanft, K., and Petzold, L. 2012 The Linear Noise Approximation is valid over limited times for any chemical system that is sufficiently large. *To appear in IET Systems Biology* ?? (??), ??,
- [15] 15. Komorowski, M., Finkenstadt, B., Harper, C., and Rand, D. 2009 Bayesian inference of biochemical kinetic parameters using the linear noise approximation. *BMC Bioinformatics* **10** (1), 343, (DOI 10.1186/1471-2105-10-343.).
- [16] 16. Robert, Christian, P. and Casella, G. 2005 *Monte Carlo Statistical Methods*, Springer Texts in Statistics, New York, USA: Springer-Verlag.
- [17] 17. Roberts, G., O., Gelman, A., and Gilks, W., R. 1997 Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms. *The Annals of Applied Probability* **7** (1), 110–120, (DOI 10.1214/aoap/1034625254.).
- [18] 18. Roberts, G., O. and Stramer, O. 2003 Langevin Diffusions and Metropolis-Hastings Algorithms. *Methodology And Computing In Applied Probability* **4** (4), 337–358, (DOI doi:10.1023/A:1023562417138.).
- [19] 19. Duane, S., Kennedy, A., B., Pendleton, J., B., and Roweth, D. 1987 Hybrid Monte Carlo. *Physics Letters B* **195** (2), 216–222, (DOI 10.1016/0370-2693(87)91197-X.).
- [20] 20. Radford, M., N. 1993 Probabilistic inference using Markov chain Monte Carlo methods, Technical Report CRG-TR-93-1 Dept. of Computer Science, University of Toronto.
- [21] 21. Roberts, G. O. and Rosenthal, J. S. 1998 Optimal Scaling of Discrete Approximations to Langevin Diffusions. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **60** (1), 255–268, (DOI 10.1111/1467-9868.00123.).

- [22] 22. Girolami, M. and Calderhead, B. 2011 Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73** (2), 123–214, (DOI 10.1111/j.1467-9868.2010.00765.x.).
- [23] 23. Ferm, L., Lötstedt, P., and Hellander, A. 2008 A Hierarchy of Approximations of the Master Equation Scaled by a Size Parameter. *Journal of Scientific Computing* **34**, 127–151, (DOI 10.1007/s10915-007-9179-z.).
- [24] 24. Tanner, M. A. and Wong, W. H. 1987 The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association* **82** (398), pp. 528–540, (DOI 10.2307/2289457.).
- [25] 25. Green, P., J. 1995 Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82** (4), 711–732, (DOI 10.1093/biomet/82.4.711.).
- [26] 26. Gillespie, D. T. 2009 Deterministic Limit of Stochastic Chemical Kinetics. *The Journal of Physical Chemistry B* **113** (6), 1640–1644, (DOI 10.1021/jp806431b.).
- [27] 27. Heron, E., A., Finkenstädt, B. and Rand, D., A. Bayesian inference for dynamic transcriptional regulation; the Hes1 system as a case study. *Bioinformatics* **23** (19), 2596–2603, (DOI 10.1093/bioinformatics/btm367.).
- [28] 28. Oksendal, B. 1992 *Stochastic differential equations: an introduction with applications*, 3rd ed., New York, NY, USA: Springer-Verlag New York, Inc..
- [29] 29. Nolan, T., Hands, R. E., and Bustin, S. A., Quantification of mRNA using real-time RT-PCR. *Nature Protocols* **1** (3), 1559–1582, (DOI 10.1038/nprot.2006.236.).
- [30] 30. Haario, H., Saksman, E., and Tamminen, E. 2005 Componentwise adaptation for high dimensional MCMC. *Computational Statistics* **20**, 265–273, (DOI 10.1007/BF02789703.).
- [31] 31. Andrieu, C. and Thoms, J. 2008 A tutorial on adaptive MCMC. *Statistics and Computing* **18** (4), 343–373, (DOI 10.1007/s11222-008-9110-y.).
- [32] 32. Kühnel, W. 2005 *Differential Geometry: Curves - Surfaces - Manifolds*, **2**, Student Mathematical Library, USA: American Mathematical Society.
- [33] 33. Haario, H., Saksman, E., and Tamminen, J. 2001 An Adaptive Metropolis algorithm. *Bernoulli* **7** (2), 223–242, (DOI 10.2307/3318737.).
- [34] 34. Amari, S.-I. and Nagaoka, H. 2000 *Methods of Information Geometry*, **191**, Translations of Mathematical Monographs, Oxford, UK: Oxford University Press.